

5

## MULTI-STANDARD TURBO INTERLEAVER USING TABLES

The present invention relates to an interleaver for a turbo encoder and decoder.

10

Wireless communication systems are widely deployed to provide various types of communications such as voice and data. One such system is wide band code division multiple access WCDMA, which has been adopted in various competing wireless communication standards, for example 3<sup>rd</sup> generation partnership project 3GPP and 3GPP2.

15

To overcome data corruption that can occur during RF transmission the different wireless communication standards typically include some form of channel coding, where one common channel coding technique is turbo coding.

20

Turbo coding involves the use of a turbo encoder for encoding a code segment (i.e. a data packet) and a turbo decoder for the decoding of the encoded code segment.

25

As shown in figure 1, a turbo encoder 100 includes two convolutional encoders 101, 102 and an interleaver 103, where the interleaver 103 shuffles (i.e. interleaves) the information bits in the packet in accordance with a specified interleaving scheme.

30

The turbo encoder 100 uses a first convolutional encoder 101 to encode information bits (i.e. systematic bits) within a code block to generate a first sequence of parity bits (i.e. parity 1 bits) in parallel to the interleaver 103 shuffling the information bits, where the shuffled information bits are encoded by a second encoder 102 to generate a second sequence of parity bits (i.e. parity 2 bits). The

- 5 information bits and the parity bits in the first and second sequence are then modulated and transmitted to a receiver.

The information bits and the first and second sequence of parity bits are received by a receiver and decoded by a turbo decoder.

10

With reference to figure 2, the turbo decoder 200 initially stores the received information bits and the parity bits in the first and second sequence in a buffer (not shown). Initially, the information bits and the first sequence of parity bits from the first convolutional encoder are retrieved from the buffer and decoded by a first decoder 201 (i.e. a first soft in soft out SISO decoder) to provide 'extrinsic' information (i.e. a-posteriori data) indicative of adjustments in the confidence in the detected values for the information bits. In one embodiment, intermediate results (i.e. a-priori) that include the extrinsic information from the first decoder 201 are then stored in the buffer in an interleaved order matching the code interleaving used at the transmitter. Alternatively, embodiment the extrinsic information from the first decoder (i.e. first SISO decoder) can be stored in the buffer in a non-interleaved order and read from the buffer in an interleaved order.

The intermediate results, the information and the second sequence of parity bits from the second encoder are retrieved from the buffer and decoded by a second decoder 202 (i.e. a second SISO decoder) to provide extrinsic information indicative of further adjustments in the confidence in the detected values for the information bits. Intermediate results that comprise the extrinsic information from the second decoder 202 (i.e. a second SISO decoder) are then stored in the buffer in a deinterleaved order complementary to the code interleaving performed at the transmitter. Alternatively, the extrinsic information can be stored in an interleaved order and read from the buffer in a deinterleaved order. The intermediate results are used in a next decoding iteration performed by the turbo decoder 200. The turbo decoder 200 performs a predetermined number of decoding iterations before producing a decision on the value of the decoded information bit.

5

Accordingly, the interleaving and deinterleaving of data is an important aspect of turbo encoding and decoding.

10 However, the interleaving and deinterleaving specifications that form part of the turbo coding/decoding process can be different for the different wireless communication standards. As such, dual standard wireless communication devices, for example a 3GPP and 3GPP2 compliant radiotelephone, typically require multiple interleavers and deinterleavers to accommodate the different standards, which can result in increased die size, power consumption and cost.

15

It is desirable to improve this situation.

In accordance with a first aspect of the present invention there is provided an interleaver for a turbo encoder or decoder according to claim 1.

20

This provides the advantage of allowing a single interleaver to support more than one wireless communication standard.

25

In accordance with a second aspect of the present invention there is provided a method for interleaving for a turbo encoder and decoder.

An embodiment of the invention will now be described, by way of example, with reference to the drawings, of which:

30

Figure 1 illustrates a known turbo encoder;

Figure 2 illustrates a known turbo decoder;

5

Figure 3 illustrates a turbo decoder according to an embodiment of the present invention;

Figure 4 illustrates an address generation unit according to an embodiment  
10 of the present invention;

Figure 5 illustrates a feature of an address generation unit according to an embodiment of the present invention;

15 Figure 3 shows a turbo decoder 300. The turbo decoder 300 includes a first SISO decoder 301, a second SISO decoder 302, a first memory module 303 and a second memory module 304.

The first SISO decoder 301 has a first maximum a posterior MAP decoding  
20 module 305 for executing a MAP algorithm, a first address generation unit AGU 306 and a first buffer 307.

The first MAP decoding module 305 is arranged to receive information bits and parity 1 bits from a transmitting unit (not shown) with the first AGU 306  
25 producing linear address sequences to allow the first MAP decoding module 305 to read intermediate results (i.e. a priori data) from the second memory module 304 and write extrinsic information (i.e. a-posterior data) generated by the first MAP decoding module 305 to the first memory module 303.

30 The second SISO decoder 302 has a second maximum a posterior MAP decoding module 308 for executing the MAP algorithm, a second address generation unit AGU 309 and a second buffer 310.

5       The second MAP decoding module 308 is arranged to receive information bits and parity 2 bits from a transmitting unit (not shown) with the second AGU 309 producing interleaving/de-interleaving address sequences to allow the second MAP decoding module 308 to read extrinsic information (i.e. a-posteriori data) from the first memory module 303 in an interleaved order and write to the second  
10   memory module 304 intermediate results (i.e. a priori data) generated by the second MAP decoding module 308 in an de-interleaved order (i.e. in the same order as the data stored in the first memory module 303).

      The first buffer 307 and second buffer 310 are arranged to temporarily store  
15   the addresses produced by the first AGU 306 and second AGU 309 respectively, thereby allowing the same address locations to be used in the first memory module 303 and second module 304 when reading and writing associated data during the same turbo decoding iteration. For example, if, during the same turbo  
20   decoding iteration, a-priori data was read from address x in the second memory module 304 then the related a-posteriori data will be written to location x in the first memory module 303.

      Although the above embodiment of a turbo decoder uses two separate SISO decoders other structures are possible. For example, as the first SISO decoder  
25   and second SISO decoder processes do not occur simultaneous a single SISO decoder could be used that is arranged to perform the first SISO decoder and second SISO decoder functions sequentially. Additionally, or alternatively, the first memory module and second memory module could be combined into a single memory module.

30

      Both the 3GPP WCDMA standard and the 3GPP2 WCDMA standard define different interleaving schemes. Even though both interleaving schemes can be categorised as having the following three stages; first, writing the information bits  
35   into a code segment row by row to a two dimensional array (i.e. Rx C array); second, interchanging the rows (i.e. inter-row permutation); and three, rearranging

5 the bits/elements within each row (i.e. intra-row permutation), where the bits are then read from the  $R \times C$  array column by column starting with the upper left most bit/element in the  $R \times C$  array; the implementation details for the 3GPP and 3GPP2 interleaving schemes are different.

10 For the 3GPP WCDMA standard the first stage of the interleaving scheme process is defined as follows:

The number of rows  $R$  in the array is determined based on the size of the code segment  $K$ , where:

15

$R = 5$ , if  $40 \leq K \leq 159$ ;

$R = 10$ , if  $160 \leq K \leq 200$  or  $481 \leq K \leq 530$ ;

$R = 20$ , for all other  $K$ .

20 The number of columns  $C$  in the array is determined based upon  $R$  and  $K$ , as follows:

$C = 53$ , if  $481 \leq K \leq 530$ ; otherwise

select a prime number  $p$  such that  $(p+1) \cdot R \geq K$ , and then

25

select  $C = \min(p-1, p, p+1)$  such that  $R \cdot C \geq K$ .

30 Once  $R$  and  $C$  have been determined for a given code segment  $K$ , the bits in the code segment are written row by row into the  $R \times C$  array, starting from row 0, column 0. If there are empty cells at the bottom of the array after the code segment has been written into the array the empty cells are padded with dummy bits.

5 For the second stage, the inter row permutation process involves permutating the rows  $R$  according to a inter row permutation sequence  $T(i)$ , where  $i$  represents the row number, which is selected from among four possible sequences  $T_A$ ,  $T_B$ ,  $T_C$  and  $T_D$ , defined by the 3GPP WCDMA standard as:

10  $T_A = (19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11);$   
 $T_B = (19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10);$   
 $T_C = (9, 8, 7, 6, 5, 4, 3, 2, 1, 0)$  and  
 $T_D = (4, 3, 2, 1, 0)$

15 where  $T_D$  is selected if  $R = 5$ ;  
 $T_C$  is selected if  $R = 10$ ;  
 $T_B$  is selected if  $(2281 \leq K \leq 2480)$  or  $(3161 \leq K \leq 3210)$   
otherwise  $T_A$  is selected.

20 For the third stage, the intra row permutation process involves permutating the column elements of the array as follows:

First a base sequence  $s$  of length  $p$  is generated where the base sequence is derived using

25  $s(j) = [v \cdot s(j-1)] \bmod p$  for  $j = 1, 2, \dots, (p-1)$  (for  $j = 0$   $s(0)$  equals 1)

where  $j$  represents the column numbers and  $v$  is a primitive root where each prime number  $p$  derived above has an associated primitive root  $v$ , as defined in the  
30 3GPP standard and as shown in table 1 below.

$p$	$v$	$p$	$v$	$p$	$v$	$p$	$v$	$p$	$v$
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

5

Table 1

Having determined the base sequence a sequence of prime numbers  $q_j$  is constructed. The prime number sequence  $q_j$  has  $R$  elements and is essentially a sequence of prime numbers that are not factors of  $(p-1)$ .

10

The prime number sequence  $q_j$  is associated with rows 0 to  $R-1$  respectively where the prime number sequence  $q_j$  elements are permuted in accordance with the appropriate inter-row permutation sequence. Accordingly, the elements of the permuted prime number sequence are determined by:

15

$$r_i = q_{\pi(i)}$$

Having determined the base sequence and permuted prime number sequence  $r_i$  the intra-row permutation sequence is determined, where for  $C = p-1$  the original bit position of the  $j$ th permuted bit of the  $i$ th row is determined from  $U_i(j)$  where:

20

$$U_i(j) = s((j \times r_i) \bmod (p-1)) - 1, \quad j = 0, 1, \dots, (p-2),$$

25

However, as stated above  $C$  can be equal to  $p-1$ ,  $p$  and  $p+1$ . Thus in the cases of  $C = p$  and  $p+1$   $U_i(j)$  needs to be extended with special cases and the '-1'

- 5 is removed from the equation. In the case of  $C = p$  then  $U_i(p-1) = 0$  and for the case of  $C = p+1$  then  $U_i(p-1) = 0$  and  $U_i(p) = p$ . Additionally, for  $K = R \cdot C$  and  $C = p+1$  the values  $U_{R-1}(p)$  and  $U_{R-1}(0)$  are exchanged.

10 The interleaved bits are read out column by column from the  $R \times C$  array from top to bottom starting from column 0, row 0.

For the 3GPP2 WCDMA standard the first stage of the interleaving scheme process is defined as follows

- 15 The number of rows  $R$  in the array is set at  $R = 32$  for a code segment  $K$  of any size.

The number of columns  $C$  in the array is determined by:

- 20  $C = 2^n$ , where  $n$  is the smallest integer such that  $2^{n+5} \geq K$   
and  
 $C = p$

- 25 Once  $R$  and  $C$  have been determined for a given code segment  $K$ , the bits in the code segment are written row by row into the  $R \times C$  array, starting from row 0, column 1. If there are empty cells at the bottom of the array after the code segment has been written into the array the empty cells are padded with dummy bits.

- 30 For the second stage, the inter row permutation process involves permutating the rows  $R$  according to the following pattern where the inter-row permutations are performed using a bit reversal rule:

5

$T = \{0, 16, 8, 24, 4, 20, 12, 28, 2, 18, 10, 26, 6, 22, 14, 30, 1, 17, 9, 25, 5, 21, 13, 29, 3, 19, 11, 27, 7, 23, 15, 31\}$

10

For the third stage, the intra row permutation process the permuted prime number sequence is determined from table 2 below:

Row	n=4	n=5	n=6	n=7	n=8	n=9	n=10
0	5	27	3	15	3	13	1
1	15	3	27	127	1	335	349
2	5	1	15	89	5	87	303
3	15	15	13	1	83	15	721
4	1	13	29	31	19	15	973
5	9	17	5	15	179	1	703
6	9	23	1	61	19	333	761
7	15	13	31	47	99	11	327
8	13	9	3	127	23	13	453
9	15	3	9	17	1	1	95
10	7	15	15	119	3	121	241
11	11	3	31	15	13	155	187
12	15	13	17	57	13	1	497
13	3	1	5	123	3	175	909
14	15	13	39	95	17	421	769
15	5	29	1	5	1	5	349
16	13	21	19	85	63	509	71
17	15	19	27	17	131	215	557
18	9	1	15	55	17	47	197
19	3	3	13	57	131	425	499
20	1	29	45	15	211	295	409
21	3	17	5	41	173	229	259
22	15	25	33	93	231	427	335
23	1	29	15	87	171	83	253
24	13	9	13	63	23	409	677
25	1	13	9	15	147	387	717
26	9	23	15	13	243	193	313
27	15	13	31	15	213	57	575
28	11	13	17	81	189	501	189
29	3	1	5	57	51	313	15
30	15	13	15	31	15	489	75
31	5	13	33	69	67	391	163

15

Table 2

The 3GPP2 intra-row permutation is described as  $x(i+1)=(x(i)+c) \bmod 2^n$ , and  $x(0)=c$  where  $c$  is the row specific value from the table. This can also be

- 5 expressed as  $x(i+1)=(i \cdot R_j) \bmod p'$ , where  $p'$  is equal to  $2^n$  for 3GPP2 (and  $p-1$  for 3GPP).  $R_j$  is the value of  $c$  for the row  $j$  and  $x(i)$  is the inter row permutation pattern.

10 Figure 4 illustrates the AGU 309, which, as stated above, is arranged to determine the interleaved addresses for both the 3GPP and 3GPP2 WCDMA standards.

The AGU 309 includes a row counter 401, a column counter 402, a first look-up table LUT 403, a second LUT 404, a third LUT 405, a fourth LUT 406, a  
15 multiply and modulo module 407, a special cases module 408, a combiner 409 and a compare module 410.

An output from the row counter 401 is coupled to the column counter 402, the first LUT 403 and the second LUT 404.

20

The first LUT 403 is arranged to store inter-row permutation data, as detailed below, with an output from the first LUT 403 being coupled to the combiner 409, which forms an inter-row permutation path.

25 The second LUT 404 is arranged to store inter-row permutation data, as detailed below, with an output from the second LUT 404 being coupled to the third LUT 405 and the special cases module 408.

30 The third LUT 405 is arranged to store permuted prime number sequences, as described below, with an output from the third LUT 405 being coupled to the multiply and modulo module 407.

- 5           An output from the column counter 402 is coupled to the multiply and modulo module 407.

The multiply and modulo module 407 has one output coupled directly to the combiner 409 and a second output coupled to the fourth LUT 406.

10

The fourth LUT 406 is arranged to store 's' the base sequence for intra-row permutations, as described below, with an output from the fourth LUT 406 being coupled to the special cases module 408.

- 15           An output from the special cases module 408, which forms an intra row permutation path, is coupled to the combiner 409 with an output from the combiner 409 being coupled to the compare module 410.

- 20           Using the inter-row permutation path and the intra-row permutation path data the compare module 410 generates interleaved addresses, as detail below, in addition to providing a clock signal for driving the row counter 401. The compare module 410 is run from a clock input signal.

- 25           A controller (not shown) (for example a DSP) is arranged to configure the AGU 309 to generate interleaved addresses in accordance with either the 3GPP or 3GPP2 WCDMA standard by initialising the relevant interleaving parameters (e.g. R, C, p) and the four LUT's in accordance with the relevant interleaving scheme criteria.

- 30           For example, if the AGU 309 is to operate according to the 3GPP WCDMA standard the controller determines the parameter values for R, C and p in accordance with the 3GPP standard and stores these values, for example in a register (not shown). Additionally, the first LUT 403, the second LUT 404, the third

- 5 LUT 405 and the fourth LUT 406 are loaded with relevant permutation pattern data according to the 3GPP WCDMA standard, as described below.

10 If the AGU 309 is to operate according to the 3GPP2 WCDMA standard the controller determines the parameter values for R, C and p in accordance with the 3GPP2 standard and stores these values. Additionally, the first LUT 403, the second LUT 404, the third LUT 405 and the fourth LUT 406 are loaded with relevant permutation pattern data according to the 3GPP2 WCDMA standard.

15 The controller can be arranged to initiate the configuration of the AGU 309 to operate in accordance with either the 3GPP or 3GPP2 standard by any suitable means, for example by user selection or by determination from received signals from a transmitting unit.

The operation of the AGU 309 will now be described.

20

The row counter 401 and column counter 402 are both initialised to reference the first RxC array element. As the 3GPP standard specifies that the first row and column is zero respectively the controller sets  $R = C = 0$ . However, as the 3GPP2 standard specifies that the first row equals 0 but the first column equals 1 the controller sets  $R = 0$  and  $C = 1$ . Typically, this operation will be implicit by starting the interleaving from a given column number.

25

For every interleaver clock cycle the row counter 401 is increased by one, to point to the next row in the array. When the row counter 401 equals R the row counter 401 initiates an increase in the column counter 402, to point to the next column in the array, and resets the row counter 401 to zero.

30

The first LUT 403 is arranged to map the permuted i row counter values (i.e. i can be regarded as the permuted row value) to the original inter-row numbers (i.e.

5 without permutation). Therefore, the  $i$  counter can be regarded as a counter that operates on the interleaved table.

Accordingly, if the AGU 309 is to be configured for operation according to the 3GPP standard the controller initiates the loading of the appropriate 3GPP inter  
10 row permutation sequence  $T(i)$ , if the AGU 309 is to be configured for operation according to the 3GPP2 standard the controller initiates the loading of the 3GPP2 inter row permutation sequence  $T$ . To avoid the use of an additional multiplier within the AGU 309 the inter row permutation data stored in the first LUT 403 is arranged to be multiplied by  $C$  prior to loading. The need to multiply the inter row  
15 permutation sequence by  $C$  arises because the inter row permutation sequence defined by the standard is written row by row, therefore the address will be  $\text{row} \times C + \text{column}$  (where  $C$  is the number of columns). For example, if  $C=10$ , the values 0-9 will be written to the first row, values 10-19 will be entered to the second row etc. The third value in the second row (row and column numbering is  
20 from zero therefore second row corresponds to  $\text{row}=1$  and the third column corresponds to  $\text{column}=2$ ) can be found from  $1 \times 10 + 2 = 12$ .

The mapped inter-row permutation address generated from the first LUT 403  
25 is output to the combiner 409.

The second LUT 404, the third LUT 405 and the multiply and modulo module 407 are arranged to map the permuted column counter values  $j$  (i.e.  $j$  can be regarded as the permuted column values) to the original intra-row numbers (i.e.  
30 without permutation).

The second LUT 404 maps the row counter values to the appropriate inter-row permuted row similar to that performed by the first LUT 403 – this is necessary to ensure that the intra-row permutation addresses correspond to the

5 same row that has been inter-row permuted by the inter row permutation path, via the first LUT 403. As with the first LUT 403, if the AGU 309 is to be configured for operation according to the 3GPP standard the controller initiates the loading of the appropriate 3GPP inter row permutation sequence  $T(i)$  into the second LUT 404, if the AGU 309 is to be configured for operation according to the 3GPP2  
10 standard the controller initiates the loading of the 3GPP2 inter row permutation sequence  $T$  into the second LUT 404, however these values do not need to be multiplied by  $C$ .

The third LUT 405 maps the row number, (i.e. the output from the second  
15 LUT 404), to the appropriate permuted prime number sequence  $r_{\text{mod}p'}$ , where  $p' = p-1$  for the 3GPP standard and  $p' = C$  for the 3GPP2 standard. Accordingly, if the AGU 309 is to be configured for operation according to the 3GPP standard the controller initiates the loading of the 3GPP permuted prime number sequence  $r_{\text{mod}p'}$  into the third LUT 405, if the AGU 309 is to be configured for operation  
20 according to the 3GPP2 standard the controller initiates the loading of the 3GPP2 permuted prime number sequence  $r_{\text{mod}p}$  into the third LUT 405. The permuted modulo prime number output is provided to the multiply and modulo module 407.

25 The multiply and modulo module 407 is arranged to multiply the received permuted modulo prime number values by the column number received from the column counter 402. Additionally, when the multiply and modulo module 407 is configured to operate in accordance with the 3GPP standard the multiply and modulo module calculates the modulo  $(p-1)$  of the received permuted modulo  
30 prime number and provides this modulo value to a first output of the multiply and modulo module 407. If, however, the multiply and modulo module 407 is configured to operate in accordance with the 3GPP2 standard the multiply and modulo module 407 calculates the modulo  $(p)$  of the resulting value and provides this value to the first output of the multiply and modulo module 407, as described  
35 below.

5           The first output of the multiply and modulo module 407 is provided as an input to the fourth LUT 406 that implements the base sequence for intra row permutations.

10           The fourth LUT 406 stores the series "s" data, as defined in 3GPP, to implement the base sequence intra row permutation. Its function is to be part of the intra-row permutation pattern as described by the equation  $U_i(j) = s((j \times r_i) \bmod (p-1)) - 1$ , where  $C = p-1$ .

15           The 3GPP2 standard does not utilize a base sequence, as such the fourth LUT 406 is populated in such a way that it is transparent, that is  $s(i)=i$ , when configured for use in the 3GPP2 standard.

20           The output from the fourth LUT 406 forms the input to the special cases module 408. The special cases module 408 makes a determination, when the AGU 309 is configured for operation in accordance with the 3GPP standard, as to whether C equals p-1, p or p+1. If C equals p the special cases module 408 sets  $U_i(p-1)$  equal to 0, if C equals p+1 the special cases module 408 sets  $U_i(p-1)$  equal to 0 and  $U_i(p)$  equal to p. Additionally, where  $K=R \times C$  and  $C=p+1$ , then the special cases module 408 is arranged to exchange  $U_{R-1}(p)$  and  $U_{R-1}(0)$ .

25

30           The output from the special cases module 408 (i.e. the intra row permuted addresses) is provided to the combiner 409 where the combiner 409 combines the inter row permutation address, via the first LUT 405, and the intra row permutation address, via the second LUT 404 and the third LUT 405, to provide interleaved addresses.

5       The multiply and modulo module 407 additionally has a second output that is configured to provide intra row permuted addresses in accordance with the 3GPP2 standard second, as described below.

10       The second output is coupled to the combiner 409, bypassing the fourth LUT 406 and the special cases module 408, where the intra row permuted addresses are combined with the respective inter row permuted addresses, provided by the inter row permutation path, to provide interleaved addresses in accordance with the 3GPP2 standard, thereby providing redundancy for 3GPP2 interleaving.

15       The interleaved addresses are provided from the combiner 409 to the compare module 410. The compare module 410 compares the received interleaved addresses with the block size (i.e. code segment size). As stated above, if a received code segment does not fully fill the RxC array the end of the array is filled with dummy bits. As such, in some cases, the generated AGU  
20       interleaved address will be higher than the block size (i.e. code segment). In these cases, the addresses that are higher than the block size correspond to the dummy bits and should be discarded. To avoid introducing irregularity into the timing of the output of the interleaved address, that can result from the discarding of the 'dummy' bits, the valid addresses are fed into a FIFO 411 within the compare  
25       module 410 where the FIFO 411 is arranged to provide valid interleaved addresses at a substantially constant rate when operating in accordance with the 3GPP standard.

30       For 3GPP2 interleaved addresses the compare module 410 compares the two address provided by the two output paths from the multiply and modulo module 407 and outputs the valid addresses, thereby ensuring for 3GPP2 interleaved addresses can be output at a constant rate through the use of the two output paths.

5 From the above description of the 3GPP interleaving scheme it can be seen that for each ten consecutive address values generated at most only two will be invalid (i.e. for every ten clock cycles there will be at least eight valid addresses). Further, two invalid addresses in two consecutive blocks of ten received addresses can only occur once for a given code segment.

10

As such, the compare module 410 includes a comparator 412 followed by the FIFO 411. The comparator 412 checks the validity of each output address by comparing the received address with the length of the received code segment, and only writes valid addresses to the FIFO 411. When the FIFO 411 is full, a "FIFO  
15 full" signal is created. This signal is used to halt the interleaver operation by stopping the clock signals being sent to the row counter 401 so the FIFO 411 will not overflow. Accordingly, the addresses can be read out of the FIFO 411 at a constant rate with the interleaving operation being stopped when necessary to avoid FIFO over flow. Typically, the clock rate applied to the row counter 401 will  
20 be faster than the clock rate used by the compare module 410 to compensate for the invalid address.

Two examples of alternatives for determining the size of the FIFO 411 and interleaver rate are:

- 25 (1) Output eight valid addresses for each 10 clock cycles. Choosing a FIFO of length of 3 ensures that the FIFO 411 will never run out of values.
- (2) Output 9 valid addresses for each 10 clock cycles. In this option a FIFO size of 4 is required to ensure the FIFO 411 will not run out of  
30 valid addresses.

Figure 5 illustrates the multiply and modulo module 407. As stated above the purpose of the multiply and modulo module 407 is to compute  $(j \cdot r_i) \bmod(p')$ , where  $p'$  is set equal to  $p-1$  when the AGU 309 is configured to operate in accordance

- 5 with the 3GPP standard and equal to C when the AGU is configured to operate in accordance with the 3GPP2 standard.

10 The multiply and modulo module 407 is arranged to utilize the fact that the AGU 309 runs sequentially on all the rows starting from column 0, when operating according to the 3GPP standard, or column 1, when operating according to the 3GPP2 standard. This allows the for an efficient recursive computation based on the equation:

$$(j*r_i)\text{mod}(p') = (((j-1)*r_i)\text{mod}(p') + (r_i)\text{mod}(p'))\text{mod}(p')$$

15

The multiply and modulo module 407 includes a fifth LUT 501, a first adder 502, a second adder 503, a third adder 504, a multiplexer 505 and a register 506 for storing  $p'$ .

- 20 The fifth LUT 501 has an input coupled to the column counter 402 and a first output coupled to the first adder 502 and a second output coupled to the second adder 503.

25 The first adder 502 has a second input for receiving permutated modulo prime sequence numbers (i.e.  $r_i\text{mod}p'$ ) from the third LUT 405 and an output that feeds back to the fifth LUT 501 and which also forms the second output for the multiply and modulo module 407, which is used when the AGU 309 is configured to operate in accordance with the 3GPP2 standard.

- 30 The second adder 503 has a second input for receiving permutated modulo prime sequence numbers from the third LUT 405 and an output that is directly coupled to one input of the multiplexer 505 and to a second input of the multiplexer 505 via the third adder 504, where the third adder 504 has an additional input for receiving  $p'$  information from the register 506.

5

The output from the multiplexer 505 provides  $(j \cdot r_i) \bmod(p')$  values where:

$$(j \cdot r_i) \bmod(p') = (((j-1) \cdot r_i) \bmod(p') + (r_i) \bmod(p')) \bmod(p')$$

10

The right hand side of the above equation defines the recursive aspect of the calculation where the first term  $((j-1) \cdot r_i) \bmod(p')$  is the value stored in the fifth LUT 501, having been input from the output of the first adder 502 in the previous recursive iteration. The second term  $(r_i) \bmod(p')$  is read from the third LUT 405, where as stated above the third LUT 405 stores  $(r_j) \bmod p'$  in order to enable this

15

recursive implementation.

The first adder 502 and second adder 503 implement the addition of the two terms from the third LUT 405 and fifth LUT 501.

20

For the 3GPP2 implementation as  $C$  is always a power of two the modulo operation is implemented by taking the appropriate number of LSB bits from the first adder 502, for the first output, and the second adder 503, for the second output.

25

For the 3GPP implementation, the modulo is calculated as follows:

Because the inputs to the second adder 503 are  $\bmod p'$  values (i.e. the inputs are smaller than  $p'$ ) the output from the second adder 503 should always be smaller than  $2p'$ . Therefore there are two options:

30

first, the second adder 503 result is smaller than  $p'$ . In this case, this is the final result and is output to the fourth LUT 406, via the multiplexer;

5

second, the second adder 503 result is bigger than  $p'$  (or equivalently, such as the second adder result minus  $p'$  is a non negative number). In this case the result is the second adder output minus  $p'$ . This subtraction is performed by the third adder 504 using the stored  $p'$  value in the register 506.

10

The multiplexer 505 chooses one of these two options according to whether the second adder result 503 minus  $p'$  is negative or not. The selected result is output to the fourth LUT 406.

15

It will be apparent to those skilled in the art that the disclosed subject matter may be modified in numerous ways and may assume many embodiments other than the preferred forms specifically set out as described above, for example the above embodiments could be arranged such that AGU could be configured to support interleaving for other wireless communication standards